Snowfall Saga

Table of contents

01 **Product Intro**
Tam Sam Som

02 **Team Intro**
Introduction

03 **Demo**
Demo of 3 features

04 **Team communication**
Tracking, communication tools, etc...

Table of contents

05 **Architecture**
Software architecture

06 **Technology**
Challenges and tools utilized

07 **Market & Context**
Tam Sam Som and competition

08 **Conclusion**
Closing remarks


01
Product Intro
Introduction to Snowfall Saga

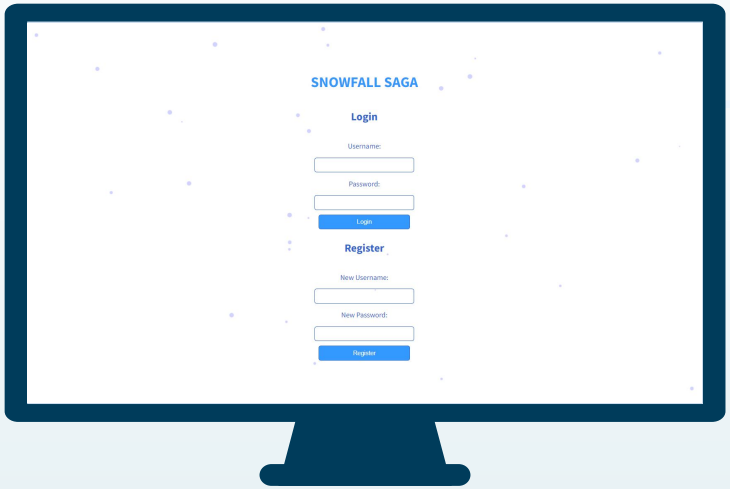
What is Snowfall Saga?

Snowfall Saga is an online game platform where players of all ages can:

- Play a variety of games with friends!
- Chat and interact with other players in an open lobby!
- Customize their character's appearance!

Our Vision

We envision Snowfall Saga as a **fun and friendly** virtual space for everyone, where players can join **quick and exciting** minigames with each other.



Our Market

TAM

(Total Addressable Market)

$78 billion

SAM

(Serviceable Addressable Market)

$12 billion

SOM

(Serviceable Obtainable Market)

$9 million


02
Team Intro
Team Introduction

Our team

Abhin Tomar

Krish Patel

Matt Taylor

Sebastian Tasson

Vaidik Patel

Jacob Youssef


03
Demo
Product showcase


04
Team Communication
Tracking, communication tools, etc...

Tracking Justification

The following are the tracking methodologies used in the project:

- Convention
- GitHub Issues
- Pull Request
- Review System

Story Tracking System

The user tracking system used in the project:

- GitHub Milestones
- GitHub Issues
- Product Backlog

Standards

The following standards were followed in the project:

- Pre-commits
- GitHub actions
- Formatted commits and Pull Requests
- Naming conventions

Communication Tools

The following are the communication tools:

- Discord
- GitHub Discussion
- Standups
- Test Meetings
- Sprint Retrospectives

Processes

We followed the following process for project:

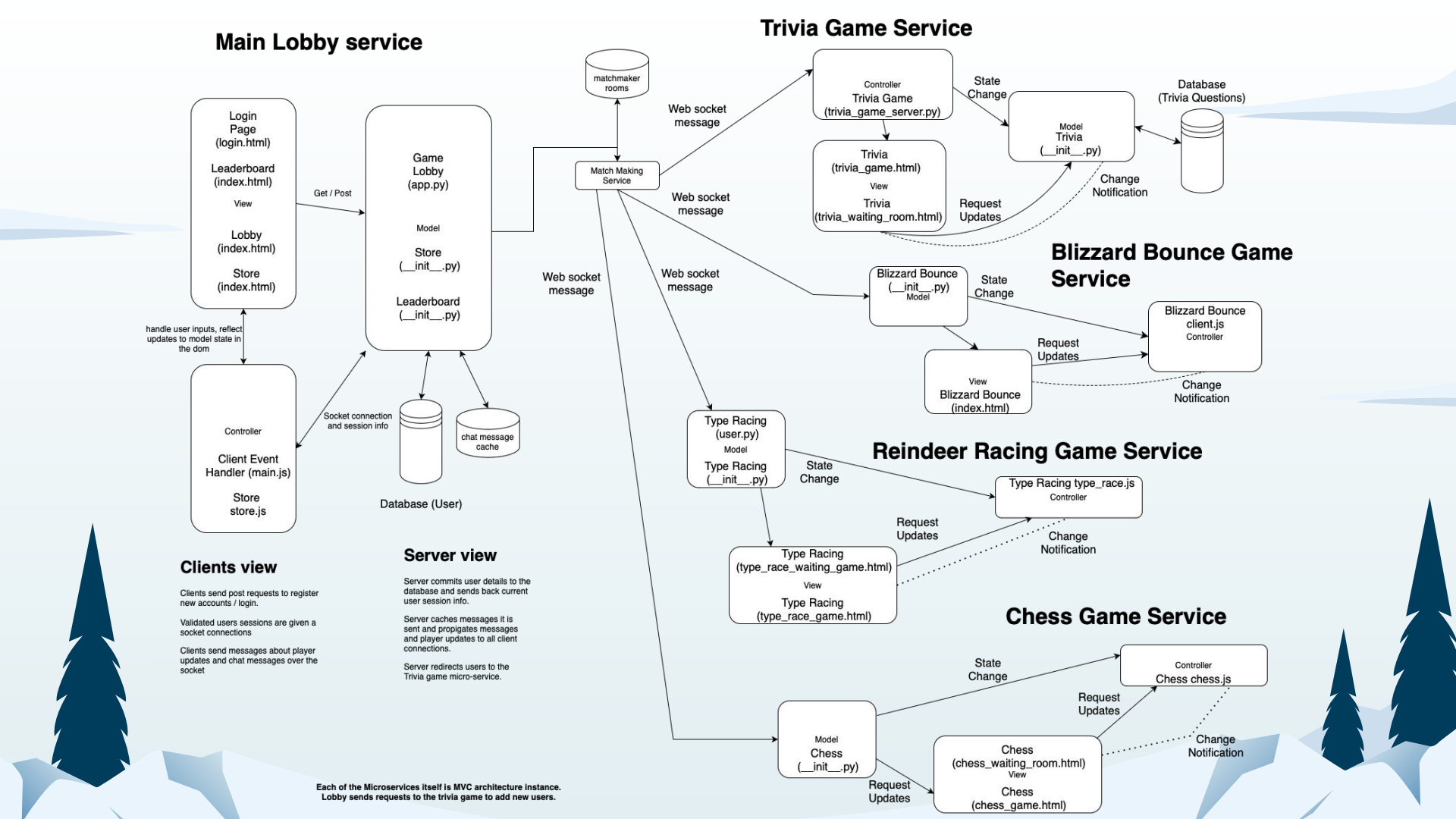
- Requirement analysis
- Planning
- Software design such as architectural design
- Software development
- Testing
- Deployment

Techniques

Following techniques were used in the project:

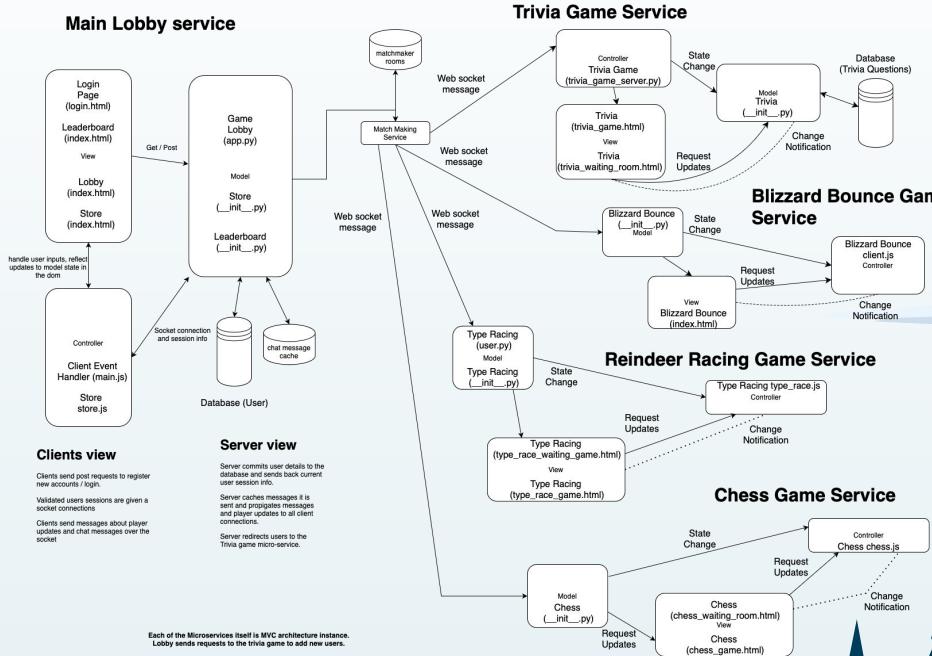
- User story prioritization technique
- Time Tracking technique
- Agile


05
Architecture
Software Architecture, components and their responsibilities


Main Lobby service
Trivia Game Service
Blizzard Bounce Game Service
Reindeer Racing Game Service
Chess Game Service
Login Page (login.html)
Leaderboard (index.html)
View
Lobby (index.html)
Store (index.html)
Get / Post
Game Lobby (app.py)
Model
Store (__init__.py)
Leaderboard (__init__.py)
handle user inputs, reflect updates to model state in the dom
Controller
Client Event Handler (main.js)
Store store.js
Socket connection and session info
chat message cache
Database (User)
matchmaker rooms
Match Making Service
Web socket message
Web socket message
Web socket message
Web socket message
Controller
Trivia Game (trivia_game_server.py)
State Change
Model
Trivia (__init__.py)
Database (Trivia Questions)
Trivia (trivia_game.html)
View
Trivia (trivia_waiting_room.html)
Request Updates
Change Notification
Blizzard Bounce (__init__.py)
Model
State Change
Blizzard Bounce client.js
Controller
Request Updates
View
Blizzard Bounce (index.html)
Change Notification
Type Racing (user.py)
Model
Type Racing (__init__.py)
State Change
Type Racing type_race.js
Controller
Request Updates
Change Notification
Type Racing (type_race_waiting_game.html)
View
Type Racing (type_race_game.html)
State Change
Controller
Chess chess.js
Request Updates
Change Notification
Model
Chess (__init__.py)
Chess (chess_waiting_room.html)
View
Chess (chess_game.html)
Request Updates
Clients view
Clients send post requests to register new accounts / login.
Validated users sessions are given a socket connections
Clients send messages about player updates and chat messages over the socket
Server view
Server commits user details to the database and sends back current user session info.
Server caches messages it is sent and propigates messages and player updates to all client connections.
Server redirects users to the Trivia game micro-service.
Each of the Microservices itself is MVC architecture instance.
Lobby sends requests to the trivia game to add new users.

High-level Overview

- Microservices architecture.
- Each microservice internally follows an MVC architecture.
- Communication between microservices is standardized and consistent.
- The main lobby sends users to the matchmaker service, which sends users to the minigames.

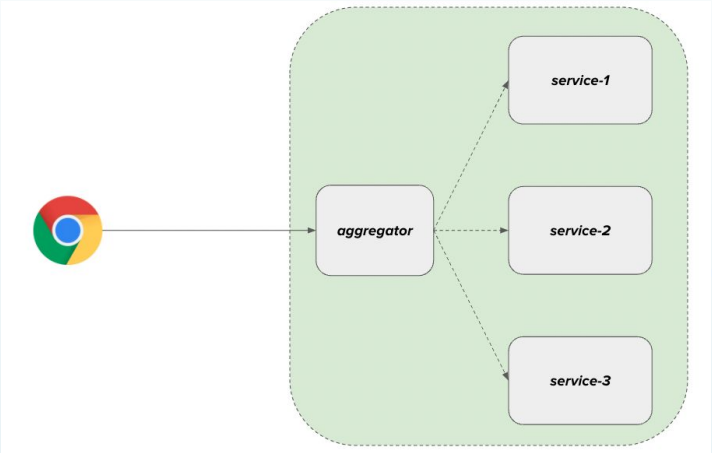


Pseudo Aggregator Pattern

Front facing lobby where users can join.

Matchmaking service that aggregates users to different games

This way we can scale to run multiple game service rooms that are aggregated through the matchmaker

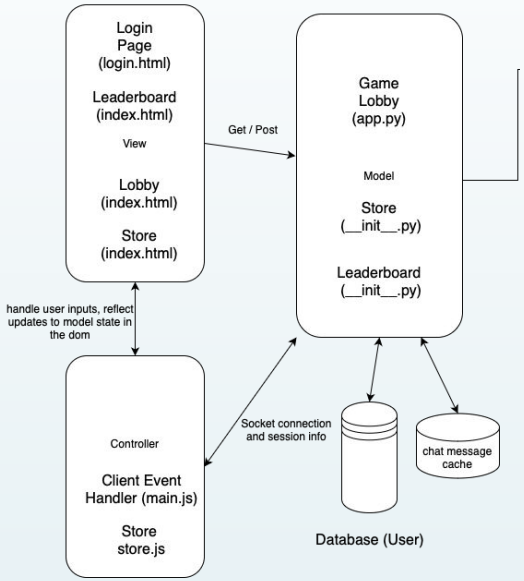


Components and Responsibilities

- In total, there are 6 microservices: The lobby service, matchmaker, Trivia game, Blizzard Bounce game, Reindeer Racing game and Chess game.
- Each service makes up a different and unique part of the user experience.
- The lobby: has various social features such as chatting and adding friends.
- The matchmaker: allows players to form groups and sends them into the minigames.
- The minigames: contain their own varied gameplay.

MVC Within each Microservice

Main Lobby service



Clients view

Clients send post requests to register new accounts / login.

Validated users sessions are given a socket connections

Clients send messages about player updates and chat messages over the socket

Server view

Server commits user details to the database and sends back current user session info.

Server caches messages it is sent and propigates messages and player updates to all client connections.

Server redirects users to the Trivia game micro-service.

Communication Between Services

- All communication from the lobby to the minigame services goes through the matchmaker first.
- All necessary information about users is stored in a current_user object from Flask, and this object is passed between each service.
- The matchmaker sends a game id (unique 6 character alphanumeric string) to the game services to identify a specific match.


06
Technology
Tech Stack and Challenges

Technologies Used

Backend	Frontend	Testing	Deployment
Flask	Javascript	PyTest	Docker
Websockets	HTML & CSS	Github Actions	Google Cloud
SQLAlchemy	Jinja Templates	Lighthouse	

Technical Challenges

- **Unfamiliarity with technologies**: Using a wide array of technologies, not everyone on the team was familiar with each
- **Choice between Framework and Library (Socketio vs Websocket)**
- **Introducing and integrating technologies:** Some of our code for the games had to be refactored for deployment with Docker
- **Testing:** Certain features, such as gameplay, were difficult to write tests for, so we had to test them manually.

07

Market & Context

The Context

What separates us from the competition, and analyzing competition

Competition: Club Penguin

Analysis:

Club Penguin passed more than 2.5 million users within a year from launch

Club Penguin was built by New Horizon Interactive and acquired by Disney.

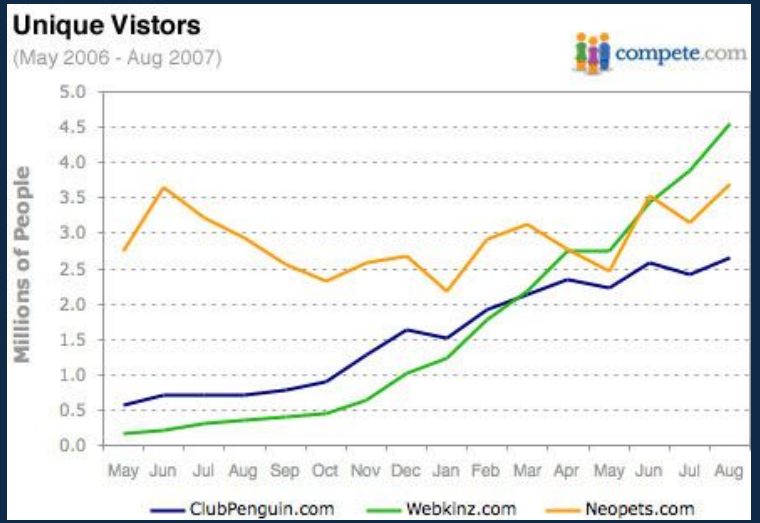
The Club Penguin game offers similar minigame services to our platform however was given a much larger budget and marketing and development team.

Club penguin charged $5.95 per month, $29.95 for six months or $57.95 for a full year for their membership subscription service model

Our niche:

– Our application has more variety in minigame choice, with more of a focus on competitive multiplayer minigames.

– Our service offers more multiplayer features within the games and allows player to interact outside the lobby.



Competition: Roblox

Analysis:
Roblox has many micro-transactions and their younger users parents have to worry about safety for their children.

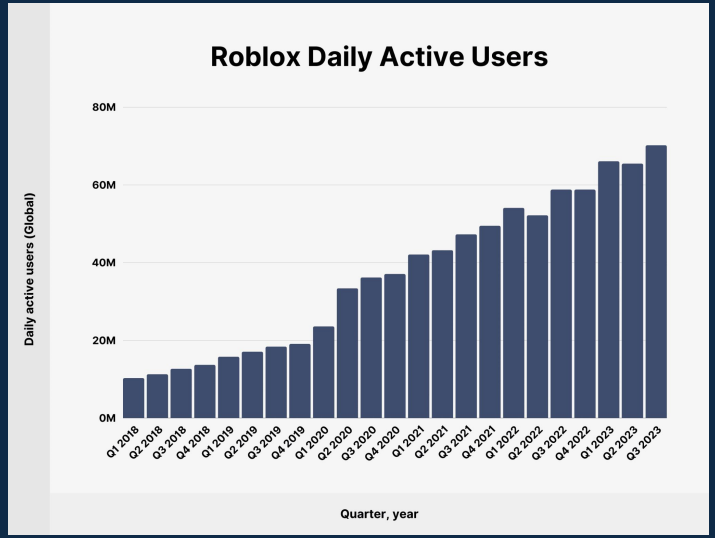
Roblox charges for ingame currency $4.99 – 400 Robux. $9.99 – 800 Robux. $19.99 – 1700 Robux.

Our niche:

– Allowing users to develop their own games in the future with in python rather than lua may bring in more creators and players

– Games are simpler in 2d easy to understand for new users.

– We have no-install step, Roblox requires users to install a roblox client to play games.



Roblox has 70.2 million daily active users and 216 million monthly active users as of 2024.

See Article

Competition: Fortnite

From their start in 2017 Fortnite had around 1 million daily active users

They were backed by Epic games a large company in the space. They also had players who were already playing on release.

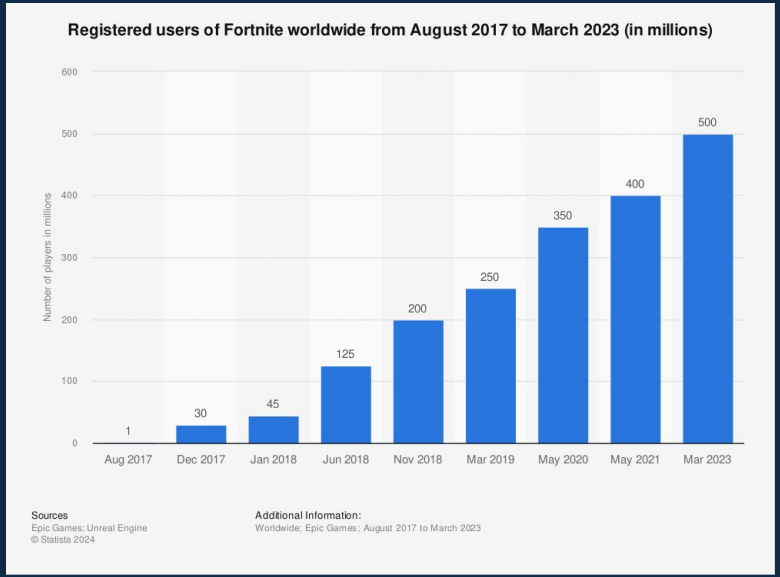
Fortnite charges $9.99 USD for the battle pass which is per season which lasts around 2 months so ~ $5.00 a month.

– Platform supports different first person shooter games using unreal engine

– First person shooter (Not very child friendly)

– Games are 3d

– Games are long and discourage players from leaving when in-game



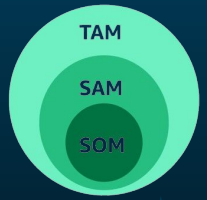
The Market

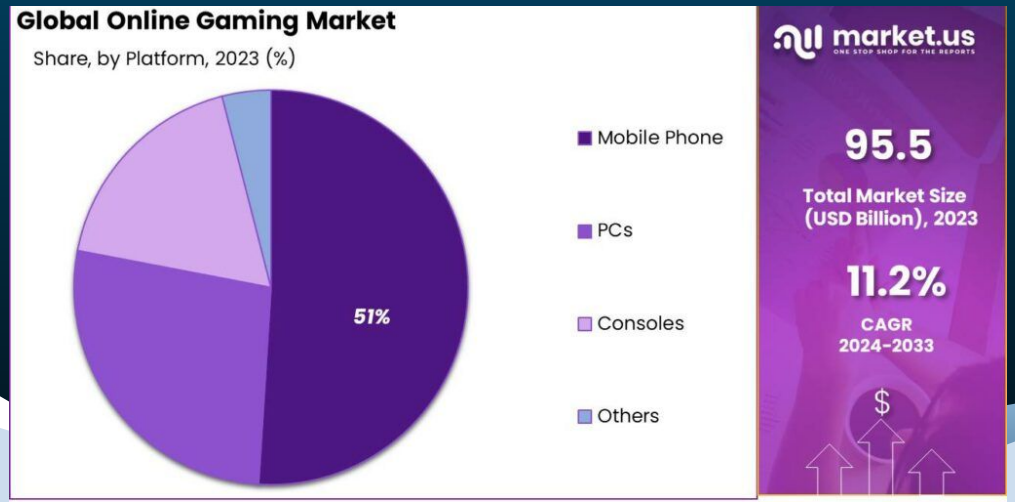
Our estimates

Total Addressable Market

For our online gaming platform hosting mini-games, TAM includes the global market for online gaming, which is significant and includes players of all ages and personal device types.

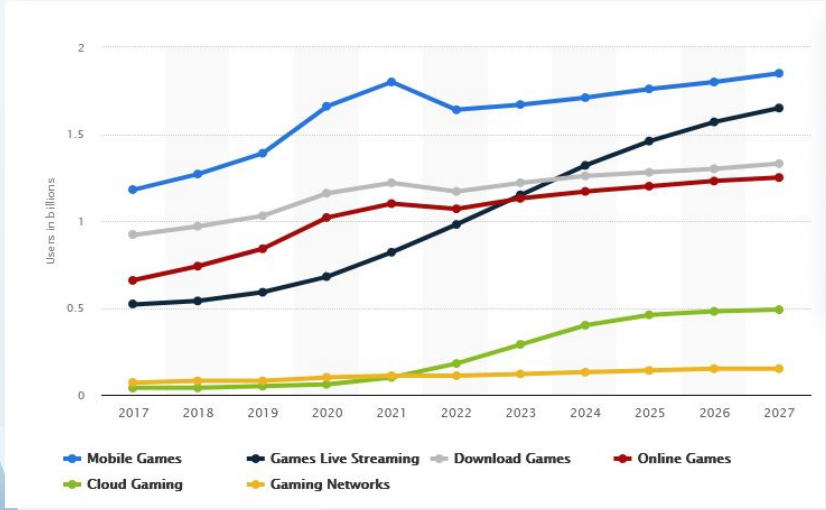
Our platform is an online web based and accessible from all devices we believe our platform can target a very large audience of users if there was no constraints.

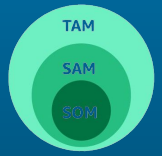




Our TAM Estimate

- 1.3 billion online video game users
- Our product costs $5.00/month -> $60.00/year
- This price is similar to our competitors: Club Penguin is $5.95/month, Fortnite is $11/2 months
- 1.3 billion * $60 = $78 billion





Serviceable Addressable Market

Sub-markets based on demographics like younger age groups, preferences in multiplayer games. Potential expansion into emerging markets where internet penetration is increasing rapidly.

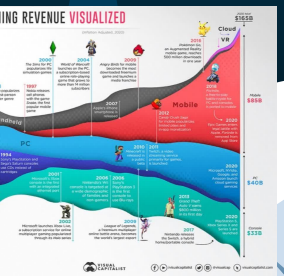
Players who prefer mini-games over other types of gaming experiences.

Users who are looking for a family-friendly gaming environment.

Those interested in social interactions within a gaming platform.

Target demographics such as children, teenagers, and students.

Gaming online: Web based platform rather than console



Our SAM Estimate

3 million

Club Penguin

total users a year after launch

200 million

Roblox

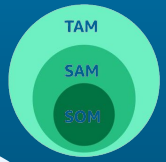
Number of monthly users

500 million

Fortnite

number of players

- Roblox is currently our closest competitor, so our SAM would be 200 million * $60 = $12 billion.

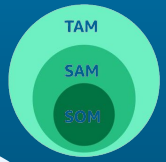


Serviceable Obtainable Market

Initial target demographics based on research and marketing efforts.

Students like to relax after school quick gaming sessions that can be interrupted without penalties to overall experience.

The geographical areas where you focus your marketing and user acquisition efforts initially. (locally within the UTM community)



Serviceable Obtainable Market

Within schools possibly high schools and university students. Games platforms like chess.com became so popular in schools that they were IP-blocked. players want a casual way to play to gain rewards without the risk of losing in-game progress

Partnerships with influencers, content creators, or gaming communities to expand your reach. (The Computer Science community loves games many CS researchers play or develop games in their limited free time our service fits their niche)

We believe we fit into some of these niche's and these users are obtainable for our platform

Our SOM Estimate

- Taking into consideration that Club Penguin got 3 million users within their first year, and we have a much smaller team of 6 developers with no marketing team, if we were to continue development and started marketing we expect our service to get ~5% of Club Penguin's numbers -> 150,000 unique visitors within a year.
- 150,000 * $60 = $9 million.

08

Conclusion

Future

We are looking at new ways to grow our services and speed up development. We plan on moving towards using protocol buffers for increasing the number of socket messages we can send. We also would like to create wrappers so python devs can make games for Snowfall Saga improving the platform into the future.